CHAOTIC SWITCH AND INFORMATION PROCESSING*

Peter Davis

ATR Optical and Radio Communications Research Laboratories,

Twin 21 MID Tower, 2-1-61 Shiromi, Higashi-ku, Osaka 540 Japan

(*presented at "Chaos and Related topics 1988" , Research Institute for Fundamental Physics, Kyoto University, December 1988)

I shall talk about my work on chaotic switch in the context of information processing. A chaotic switch is a way of controlling transitions between attracting basins in a dynamical system, and as such is a way of accessing information stored in the form of attractors. I begin by comparing direct and stochastic search methods for locating basins. I then introduce the chaotic switch method for doing stochastic search, with an example in a delayed-feedback model [1]. Finally, I will explain how the chaotic switch method might be used to control complex memory states .

A fundamental task in information processing is to produce an output , O=B, corresponding to an input, I=A, by using the input to access a memory state which determines pairing of inputs with appropriate outputs (I,O)=(A,B). However, even if the input is simple, the memory state might be complex, ie. the memory might be a complex dynamical system. Let us consider a memory system (figure 1) consisting of an underlying complex dynamics X, which is interfaced to the exterior via a mechanism which extracts lower dimensional features, R. From the exterior we see X only as its image in R space.

As an example, let us consider the delayed-feedback dynamical system (DFS) defined by a delay-differential equation of motion [3], [4].

$$dx(t)dt = -x(t) + \mu\,\pi\,(1 + \sin x(t - t_r))$$

Because of the feedback delay of t_r , the DFS produces a pattern of length t_r , $X=(x(t),...,x(t+t_r))$, which evolves in time. I emphasize that the primary signal space is an infinite-dimensional space, in which a point correponds to a signal X of length t_r. For long enough t_r , X exhibits oscillations and chaos. From the signal X, features R can be extracted with a feature detector in the following way. First we input X to a multiple channel pulse generator which, given a threshold value and a pulse width for each channel, generates a set of secondary signals S. I use the average pulse frequency of S_i as feature R_i. With this feature detector we can get a constant vector R if the primary signal is periodic.

What happens when we try to access the high dimensional oscillation states X with one of the features R? In general, we can run into problems if we use a direct access method. By *direct access* I mean putting into the DFS a signal $X_0=(x(0),...,x(t_r))$ which determines an initial condition for the delay-differential equation of motion. If the only information we use is $R_1=A$, we must choose randomly an X_0 from the subset X_A of X which have feature $R_1=A$. If all such X_0 lie in the basin of attraction of a stable oscillation with $R=(A,B)$, then the asymptotic output $R_2=B$ corresponding to input $R_1=A$ will be meaningful. However, if some X_0 in X_A lie in the basin of attraction of a state for which $R_1 \neq A$, then the asymptotic output might not be meaningful. Also, if as in the third case, there are multiple attractors for which $R_1=A$, we will be choosing one of them at random. Is this an appropriate answer not to get information from the other satisfactory memory states?

Thus problems arise when the feature vector doesn't have enough information to access the memory state directly. One way of finding something when you don't have enough information to locate it directly is to search for it by trial and error - allow some randomness in the motion through phase space, monitor the dynamical state , comparing features with the required features, and decrease the randomness when you find a satisfactory state, trapping the system near at or near the satisfactory state, say the one with $R_1 = A$. This is *stochastic search access* of the state. How do we introduce the stochasticity into the system? We could consider thermal noise ([5], [6], [7]) or chaos. I will show how we can do a search with chaos.

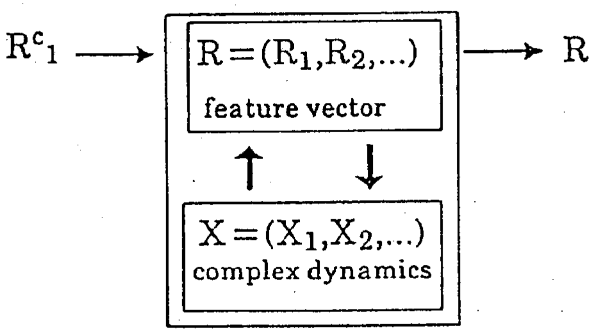

Figure 1. Access of complex states X via simple features R.

Specifically, I will show how we can do a *chaotic search* and *chaotic switch* in the delayed-feedback model. The DFS system and the feature detector giving the feature vector R are as before. To this we add a comparison with an input feature value R^c and feedback the error to determine the parameter μ. (figure 2)

The parameter μ is a bifurcation parameter for the DFS. Figure 3 shows schematically the bifurcation of two oscillation patterns α and β. Increasing μ results in cascaded bifurcation of oscillations, the onset of chaos, and an inverse bifurcation cascade in which basins merge to create intermittent chaos between oscillation patterns. ie. we can cause chaotic transitions between our two oscillation patterns α and β by increasing the parameter μ.

For our chaotic search we vary μ according to the control rule shown in figure 4. For the current μ we observe feature R (over time T) to determine a point in the

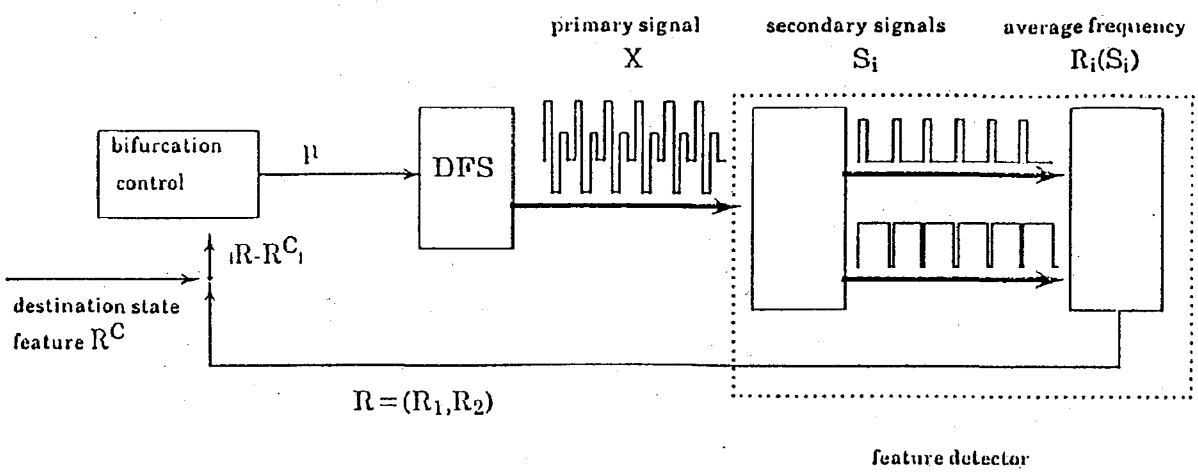

Figure 2. Scheme for chaotic search and switch in delayed-feedback system(DFS).

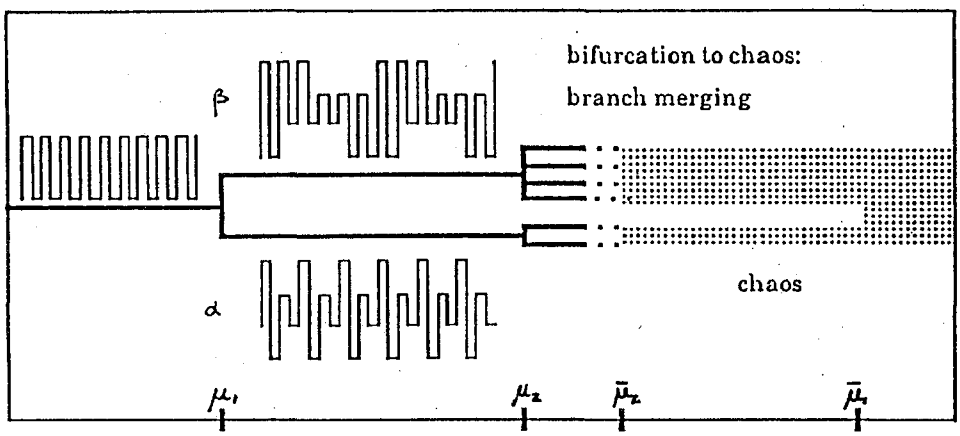

Figure 3. Bifurcation of oscillations α and β with increase of parameter μ.

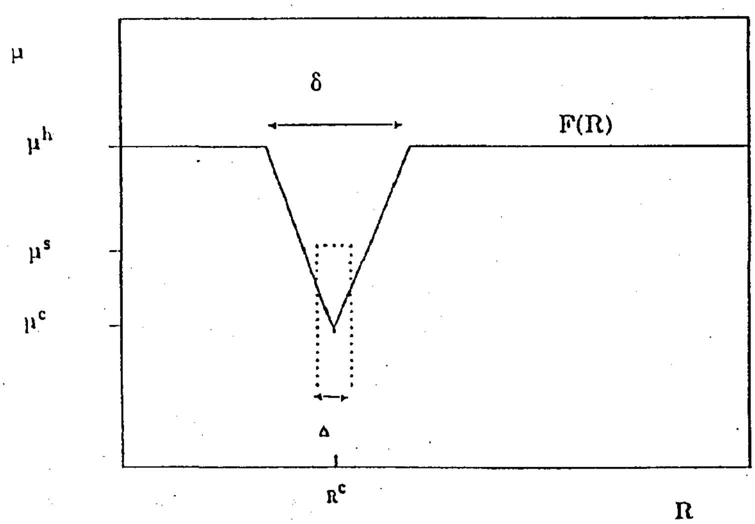

Figure 4. Attracting surface F(R) in control space (μ,R).

control space (μ,R). If the feature R is sufficiently near the input feature vector R^c, and μ is sufficiently small (ie. in the dotted region) we don't change μ. Otherwise we move μ toward the surface F(R). The surface F(R) is defined by the value of μ^h, μ^c and δ. Assume μ^h is above the onset of intermittent chaos at μ^m where the system makes transitions between basins and R makes transitions between corresponding values of R. If R^c is near one of these values, the system may be trapped in the dotted domain.

Figure 5 shows an example of a switch from pattern α to pattern β. At "switch on" we change R^c from the value for α to the value for β. The system quickly

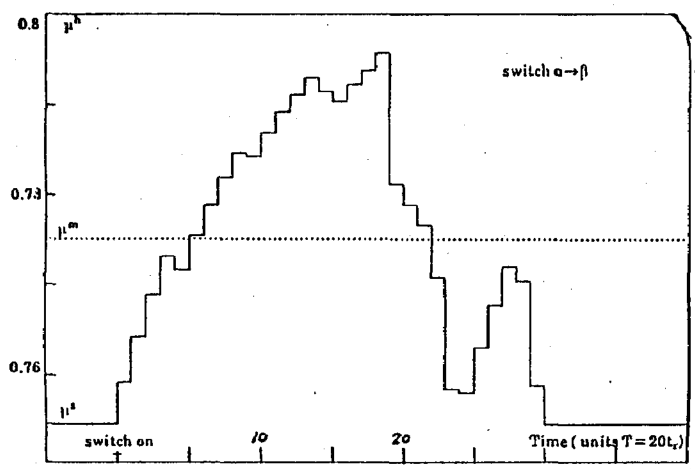

Figure 5. Example of change of μ with time during switch from α to β.

climbs up into chaos, wanders there for a while, and then "relaxes", ("cools", "is trapped") into the stable β oscillation. The switch took about $30T = 600t_r$. We can tune the control rule parameters to optimize the average switch time. The switch time depends on a balance between switch parameters such as μ^h, μ^c, δ, Δ, R^c, T, and bifurcation structure of the DFS. A switch can be reliable in the sense that it is highly likely it will be completed within a certain time.

More generally speaking, (by raising μ^c) this scheme works for switching at the merging of the branches corresponding to α and β at $\mu_1 = \mu^m$, even though the states are still chaotic below there, because the chaos there does not affect the feature R. Even more generally speaking (by further raising μ^c) this switch scheme can be used even above the basin merging point to control the chaos, as observed for example in the probability distribution of R.

This suggests we can use the chaotic switch control principle to realize a meaningful chaotic memory. As we have seen in the example, if we insist on convergence we can use the chaotic switch scheme as a stochastic means to resolve the first problem of direct access associated with wrong answers due to insufficient input information. However, if we don't insist on complete convergence in the second problem case of multiplicity of satisfactory answers, we may realize a new type of memory. The principle I advocate for such a memory is that it should not converge more than necessary - I call this a maximum dynamical entropy principle. Convergence is caused by the existence of states with features corresponding to input features. For example for input feature $R^c{}_1$ in the figure 6. The way this input via chaotic switch affects the memory state is the answer to the input. eg. if convergence of the input feature

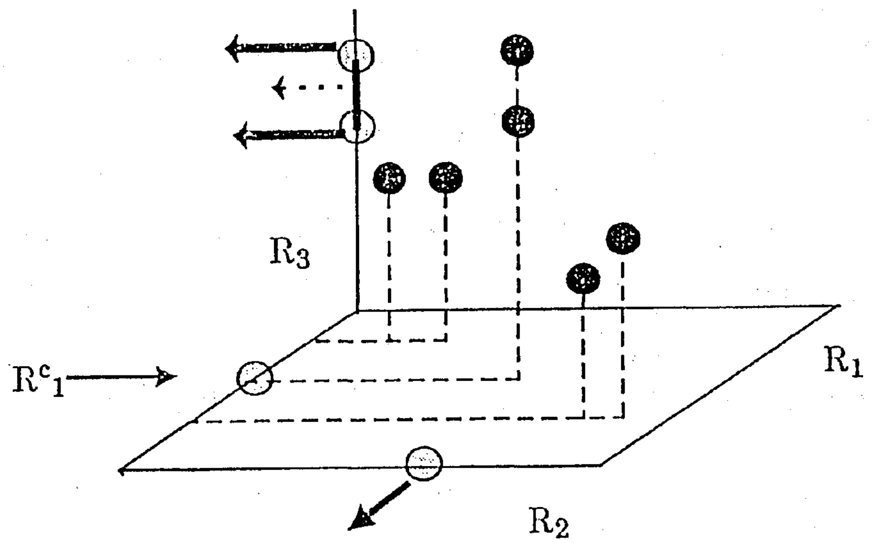

Figure 6. Schematic of search access of chaotic memory.

R_1 results in convergence of another feature, R_2, the memory has an association between these features. If there is degeneracy in a feature, such as R_3, with respect to the input R_1, the output may be intermittent, reflecting this ambiguity. Thus there can be a well-defined correspondence between an input feature, and a chaotic memory state. The nature of this correspondence is worth studying. At the least it provides ouput indicating ambiguity, with intermittent chaos between and around the ambiguous features.

(*Appendix*) I propose that the principle of chaotic switch suggests a general paradigm for chaotic control and logic - a paradigm relating input and output which would not necessarily require the system to converge to a non-chaotic state. In this paradigm the input of a set of features, possibly compounded in AND/OR combinations, results in a reshaping of the complex dynamics X(t) due to the constraining of the corresponding features. The output of this reshaped state X or its projection onto feature space R would give a type of membership state corresponding to the input.

REFERENCES

[1] Peter Davis "Chaotic switch in a delayed-feedback system" submitted for publication 1988.

[2] K. Tomita, "The significance of the concept 'chaos'", Prog. Theor.Phys. Supplement 79 (1984) 1

[3] J. S. Nicolis "Chaos and Information Processing" Rep. Prog. Phys. 49 (1986) 1109

[4] K. Ikeda and K. Matsumoto "High-dimensional chaos in delayed-feedback model" Physica 29D (1987) 194

[5] H. Gibbs, "Optical Bistability: Controlling light with light" (Academic Press Inc., Orlando, 1985)

[6] D. H. Ackley "A connectionist machine for genetic hillclimbing", (Kluwer Academic Publishers, Boston, 1987)

[7] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing", Science 220 (1983) 671

[8] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6 (1984) 721